CLAIMS:

We claim:

1. A system for improving TCP throughput over lossy communication links without affecting performance over non-lossy links comprising:
   - means for determining lookahead-loss which is the number of lost packets in a given **loss-window.**
   - means for using **said loss-window** and said lookahead loss to detect congestion in said communication links, and
   - means for controlling transmission under congestion conditions as well as under normal conditions.

2. The system as claimed in claim 1, wherein said means for determining lookahead-loss is a mechanism for identifying the number of packets transmitted by the sender in said **loss-window**, for which either of the following conditions is true:
   - sender has received at least max-**dupacks** (an appropriately selected number, typically three) duplicate cumulative acknowledgements,
   - sender has neither received **acknowledgement** nor selective **acknowledgement** for said packets, while it has received selective acknowledgements for at least max-dupsacks (an appropriately selected number, typically three) packets with higher sequence numbers.

3. The system as claimed in claim 1, wherein said means for detecting congestion is a mechanism for identifying when the number of packets lost in a loss-window is greater than an appropriately selected preset number.

4. The system as claimed in claim 1, wherein said means for controlling transmission is a TCP k-SACK protocol which is a modification of the fast retransmit algorithm of the basic congestion control algorithm of TCP to include:

- entering a 'halt growth phase' whenever lookahead loss is greater than zero and congestion is not detected,
- entering a 'k-recovery phase' whenever the congestion is detected.

5. The system as claimed in claim 4, wherein said 'halt growth phase', the sender freezes the congestion window and maintains it in that state.

6. The system as claimed in claim 4, wherein said entry into **'k-recovery phase'** reduces the congestion window to half its original size, while the slow-start threshold is reduced to half only on the first occasion of entry into the k-recovery phase during a packet loss recovery cycle.

7. The system as claimed in claim 4, further including:

- "Post Recovery" wherein the sender continues in congestion avoidance or slow-start phase at the end of the fast recovery phase,
- more accurate estimation of pipe size using the received selective acknowledgement (SACK) data, and
- use of said accurate pipe size information for controlling window

inflation and deflation thereby allowing **quicker retransmission** of lost packets and resulting faster recovery.

8. A method for improving TCP throughput over lossy communication links without affecting performance over non-lossy links comprising:
   - determining lookahead-loss which is the number of lost packets in a given **loss-window.**
   - using said **loss-window** and said lookahead loss to detect congestion in said communication links, and
   - controlling transmission under congestion conditions as well as under normal conditions.

9. The method as claimed in claim 8 wherein said determining of lookahead-loss is for identifying the number of packets transmitted by the sender in said **loss-window**, for which either of the following conditions are true:
   - sender has received at least max-**dupacks** (an appropriately selected number, typically three) duplicate cumulative acknowledgements,
   - sender has neither received **acknowledgement** nor selective **acknowledgemnt** for said packets, while it has received selective acknowledgements for at least max-dupsacks (an appropriately selected number, typically three) packets with higher sequence numbers.

10. The method as claimed in claim 8, wherein said detecting of congestion is for identifying when the number of packets lost in a loss-window is greater than an appropriately selected preset number.

11. The method as claimed in claim 8, wherein said controlling of transmission is a TCP k-SACK protocol which is a modification of the fast retransmit algorithm of the basic congestion control algorithm of TCP to include

- entering a 'halt growth phase' whenever lookahead loss is greater than zero and congestion is not detected,
- entering a 'k-recovery phase' whenever the congestion is detected.

12. The method as claimed in claim 11, wherein during said 'halt growth phase', the sender freezes the congestion window and maintains it in that state.

13. The method as claimed in claim 11, wherein during said 'k-recovery phase' reduces the congestion window to half its original size, while the slow-start threshold is reduced to half only on the first occasion of entry into the k-recovery phase during a packet loss recovery cycle.

14. The method as claimed in claim 11, further including:
- "Post Recovery" wherein the sender continues in congestion avoidance or slow start phase at the end of the fast recovery phase,
- more accurate estimation of pipe size using the received selective acknowledgement (SACK) data,
- use of said accurate pipe size information for controlling window inflation and deflation thereby allowing early retransmit of lost packets and resulting faster recovery.

15 A computer program product comprising computer readable program code stored on computer readable storage medium embodied therein for improving TCP throughput over lossy communication links without affecting performance over non-lossy links comprising:

- computer readable program code means configured for determining lookahead-loss which is the number of lost packets in a given **loss-window,**
- computer readable program code means configured for using said **loss-window** and said lookahead loss to detect congestion in said communication links, and
- computer readable program code means configured for controlling transmission under congestion conditions as well as under normal conditions.

16. The computer program product as claimed in claim 15, wherein said computer readable program code means configured for determining lookahead-loss is a mechanism for identifying the number of packets transmitted by the sender in **said loss-window**, for which either of the following conditions is true:

- sender has received at least **max-dupacks** (an appropriately selected number, typically three) duplicate cumulative acknowledgements,
- sender has neither received **acknowledgement** nor selective **acknowledgement** for said packets, while it has received selective acknowledgements for at least max-dupsacks (an appropriately selected number, typically three) packets with higher sequence numbers.

17. The computer program product as claimed in claim 15, wherein said computer readable program code means configured for detecting congestion is a mechanism for identifying when the number of packets lost in a loss-window is greater than an appropriately selected preset number.

18. The computer program product as claimed in claim 15, wherein said computer readable program code means configured for controlling transmission is a TCP k-SACK protocol which is a modification of the fast retransmit algorithm of the basic congestion control algorithm of TCP to include:

- entering a 'halt growth phase' whenever lookahead loss is greater than zero and congestion is not detected,
- entering a 'k-recovery phase' whenever the congestion is detected.

19. The computer program product as claimed in claim 18, wherein during said 'halt growth phase', the sender freezes the congestion window and maintains it in that state.

20. The computer program product as claimed in claim 18, wherein during said 'k-recovery phase' reduces the congestion window to half its original size, while the slow-start threshold is reduced to half only on the first occasion of entry into the k-recovery phase during a packet loss recovery cycle.

21. The computer program product as claimed in claim 18, further including:

- "Post Recovery" wherein the sender continues in congestion avoidance

or slow start phase at the end of the fast recovery phase,

- more accurate estimation of pipe size using the received selective acknowledgement (SACK) data,
- use of said accurate pipe size information for controlling window inflation and deflation thereby allowing early retransmit of lost packets and resulting faster recovery.